

|              |     |          |     |          |     |              |            |            |
|--------------|-----|----------|-----|----------|-----|--------------|------------|------------|
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTTTTTTTTTTT |            | SSSSSSSSSS |
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTTTTTTTTTTT |            | SSSSSSSSSS |
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTTTTTTTTTTT |            | SSSSSSSSSS |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBBBBBBBBBBB |     | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBBBBBBBBBBB |     | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBBBBBBBBBBB |     | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBB          | BBB | 000      | 000 | 000      | 000 | TTT          | SSS        |            |
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTT          | SSSSSSSSSS |            |
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTT          | SSSSSSSSSS |            |
| BBBBBBBBBBBB |     | 00000000 |     | 00000000 |     | TTT          | SSSSSSSSSS |            |

```
BBBBBBBBB  TTTTTTTTT  MM      MM  EEEEEEEEE  MM      MM  77777777  555555555  000000
BBBBBBBBB  TTTTTTTTT  MM      MM  EEEEEEEEE  MM      MM  77777777  555555555  000000
BB      BB  TT      TT  MMMM  MMMM  EE      EE  MMMM  MMMM  77      55      00      00
BB      BB  TT      TT  MMMM  MMMM  EE      EE  MMMM  MMMM  77      55      00      00
BB      BB  TT      TT  MM  MM  MM  EE      EE  MM  MM  MM  77      555555  00      0000
BB      BB  TT      TT  MM  MM  MM  EE      EE  MM  MM  MM  77      555555  00      0000
BBBBBBBBB  TT      TT  MM      MM  EEEEEEEEE  MM      MM  77      77      55      55      00      00
BBBBBBBBB  TT      TT  MM      MM  EEEEEEEEE  MM      MM  77      77      55      55      00      00
BB      BB  TT      TT  MM      MM  EE      EE  MM      MM  77      77      55      55      0000  00
BB      BB  TT      TT  MM      MM  EE      EE  MM      MM  77      77      55      55      0000  00
BB      BB  TT      TT  MM      MM  EE      EE  MM      MM  77      77      55      55      0000  00
BB      BB  TT      TT  MM      MM  EE      EE  MM      MM  77      77      55      55      0000  00
BBBBBBBBB  TT      TT  MM      MM  EEEEEEEEE  MM      MM  77      77      55      55      000000  00
BBBBBBBBB  TT      TT  MM      MM  EEEEEEEEE  MM      MM  77      77      55      55      000000  00
                                         .....
                                         .....
                                         .....
                                         .....

LL          IIIIIII  SSSSSSSS
LL          IIIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLLL IIIIIII  SSSSSSSS
LLLLLLLLLLL IIIIIII  SSSSSSSS
```



|     |     |                                      |
|-----|-----|--------------------------------------|
| (2) | 131 | Declarations                         |
| (3) | 157 | CHECKMEM_750, Identify 11/750 memory |

```
0000 1 .TITLE BTMEM750 - Configure and Test 11/750 Memory
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28 ++
0000 29
0000 30 FACILITY:
0000 31
0000 32 Linked with VMB.EXE - part of the
0000 33 bootstrap module for VAX 11/750 hardware.
0000 34
0000 35 ENVIRONMENT:
0000 36
0000 37 Runs at IPL 31, kernel mode, memory management is OFF, IS=1
0000 38 (running on interrupt stack), and code must be PIC.
0000 39
0000 40
0000 41 FUNCTIONAL DESCRIPTION:
0000 42
0000 43 This routine is 11/750 specific and
0000 44 determines how many memory controllers are on the system,
0000 45 where they are, how much memory they control, which pages
0000 46 of that memory are present (and good). Then the routines
0000 47 set bits in the PFN bitmap to represent each present (and
0000 48 good) page of memory.
0000 49
0000 50 As a side effect, the routines store the type of adapter located
0000 51 at each bus slot in the RPB.
0000 52
0000 53 INPUTS:
0000 54
0000 55 R5 - address of 1st RPB configuration code field
0000 56 R7 - address of the SCB
0000 57 R11 - address of the RPB
```



```
0000 58 :  
0000 59 : IMPLICIT INPUTS:  
0000 60 :  
0000 61 : The positions on the 11/750 system bus are called slots, and are  
0000 62 : identified by slot numbers 16-31. Slots 16-25 are called fixed  
0000 63 : slots. If an adapter is present in a fixed slot, the adapter  
0000 64 : must be of a predefined type. For example,  
0000 65 :  
0000 66 : slot 16 = memory controller 0 = ^XF20000  
0000 67 : slot 20 = MASSBUS adapter 0 = ^XF28000  
0000 68 : slot 24 = UNIBUS adapter 0 = ^XF30000  
0000 69 :  
0000 70 : Adapters at fixed slots do not have configuration registers.  
0000 71 : This routine determines their presence by reading the first  
0000 72 : longword of the slot and not having the read result in a  
0000 73 : non-existent memory machine check. The data resulting from the  
0000 74 : read will be garbage.  
0000 75 :  
0000 76 : 10 fixed system bus slots are currently defined for the 11/750:  
0000 77 :  
0000 78 : 4 memory controllers, starting at ^XF20000  
0000 79 : 4 MASSBUS adapters, starting at ^XF28000  
0000 80 : 2 UNIBUS adapters, starting at ^XF30000  
0000 81 :  
0000 82 : The other 6 system bus slots are floating slots. If an adapter  
0000 83 : is present, that adapter must have a configuration register as  
0000 84 : the first location in the slot's address space. The register  
0000 85 : must contain the adapter type in the low byte.  
0000 86 :  
0000 87 : The 11/750 currently supports only 1 memory controller, located  
0000 88 : at ^XF20000. The 11/750 does not have interleaved memory. This  
0000 89 : routine tests all controller slots as though memory could exist  
0000 90 : at them.  
0000 91 :  
0000 92 : Memory controller registers also contain the starting page  
0000 93 : number / 128. This routine determines where the end of memory  
0000 94 : on a memory controller is by analyzing the memory present map  
0000 95 : in the third memory controller register.  
0000 96 :  
0000 97 : OUTPUTS:  
0000 98 :  
0000 99 : R7, R8, R11, and SP are preserved  
0000 100 : All others (including AP and FP) are altered  
0000 101 :  
0000 102 : IMPLICIT OUTPUTS:  
0000 103 :  
0000 104 : The PFN bitmap is modified to describe all of physical memory.  
0000 105 :  
0000 106 : RPB$$_PFNCNT stores the number of pages of physical memory.  
0000 107 :  
0000 108 : All single parity errors in memory are cleared.  
0000 109 :  
0000 110 : RPB$$_CONFREG describes each NEXUS on the system bus with an  
0000 111 : adapter type code.  
0000 112 :  
0000 113 : AUTHOR:  
0000 114 :
```

```
0000 115 : C. A. Samuelson, creation date 24-April-1981
0000 116 :
0000 117 : REVISION HISTORY:
0000 118 :
0000 119 : V03-002 TCM0003 Trudy C. Matthews 27-Apr-1983
0000 120 : Change sense of CRDTEST flag from an enable to an inhibit, i.e.
0000 121 : remove pages with CRD errors by default.
0000 122 :
0000 123 : V03-001 TCM0002 Trudy C. Matthews 26-Jan-1983
0000 124 : Add support for RPB$V_CRDTEST flag, which specifies that
0000 125 : pages with CRD errors be removed.
0000 126 :
0000 127 : V03-001 KDM0078 Kathleen D. Morse 15-Mar-1982
0000 128 : Clear all MA780-specific boot flags.
0000 129 :--
```



```
0000 131      .SBTTL  Declarations
0000 132
0000 133      .DEFAULT DISPLACEMENT, WORD
0000 134
0000 135  ::
0000 136  :: Macros to describe VMS data structures
0000 137  ::
0000 138
0000 139      $DMPDEF      ; System dump file header definitions
0000 140      $IO750DEF   ; 11/750 definitions
0000 141      $NDTDEF     ; Nexus device types
0000 142      $PRDEF      ; Processor registers
0000 143      $RPBDEF     ; Restart parameter block
0000 144      $SUBDEF     ; 11/750 UNIBUS adapter
0000 145
0000 146  ::
0000 147  :: Macros
0000 148  ::
0000 149
0000 150      .MACRO  ERROR,STR      ; Outputs an error string to the
0000 151      BSBW      ERRROUT      ; console terminal.
0000 152      .ASCIZ  STR
0000 153      .ENDM    ERROR
0000 154
00000000 155      .PSECT  YBTMEM, LONG
```

```
0000 157 .SBTTL CHECKMEM_750, Identify 11/750 memory
0000 158
0000 159 :++
0000 160 :
0000 161 : CHECKMEM_750, Locate and test memory for 11/750
0000 162 :
0000 163 :--
0000 164 :
0000 165 :
0000 166 : The table that follows identifies the adapter type codes of adapters
0000 167 : located in fixed 750 SBI slots. The last 6 slots contain zeroes to
0000 168 : indicate floating slots.
0000 169 :
0000 170 :
0000 171 ADAP_TYPE 750: : Adapter type table for 11/750
10 0000 172 .BYTE NDT$_MEM16NI : Memory controller 0.
40 0001 173 .BYTE NDT$_MPM0 : Multiport memory 0.
41 0002 174 .BYTE NDT$_MPM1 : Multiport memory 1.
42 0003 175 .BYTE NDT$_MPM2 : Multiport memory 2.
20 0004 176 .BYTE NDT$_MB : MASSBUS adapter.
20 0005 177 .BYTE NDT$_MB : MASSBUS adapter.
20 0006 178 .BYTE NDT$_MB : MASSBUS adapter.
20 0007 179 .BYTE NDT$_MB : MASSBUS adapter.
28 0008 180 .BYTE NDT$_UB0 : UNIBUS adapter 0.
29 0009 181 .BYTE NDT$_UB1 : UNIBUS adapter 1.
00 000A 182 .BYTE 0 : Floating slot.
00 000B 183 .BYTE 0 : Floating slot.
00 000C 184 .BYTE 0 : Floating slot.
00 000D 185 .BYTE 0 : Floating slot.
00 000E 186 .BYTE 0 : Floating slot.
00 000F 187 .BYTE 0 : Floating slot.
0010 188
0010 189
0010 190 :
0010 191 : Get address of appropriate fixed slot assignment table for each
0010 192 : CPU:
0010 193 :
0010 194 :
52 ED AF 9E 0010 195 CHECKMEM 750:: : Entry for 11/750.
0010 196 MOVAB ADAP_TYPE_750,R2 : Get 11/750 fixed slot assignment
0014 197 : table
0014 198 :
0014 199 : Start testing slot positions to find adapters. First save the stack
0014 200 : position so it can be restored after a machine check.
0014 201 :
0014 202 :
5C 10 9A 0014 203 INIT_SEARCH: : Start searching for adapters
5D 5E D0 0014 204 MOVZBL #I0750$AL_NNEX,AP : Set up NEXUS loop counter.
0017 205 MOVL SP,FP : Save current top of stack.
001A 206
001A 207 :
001A 208 : Set up the physical address of the 1st slot on the system bus and
001A 209 : the address of the adapter type table.
001A 210 :
54 00F20000 9F 9E 001A 211
0021 212 MOVAB a#I0750$AL_IOBASE,R4 : Get address of 1st slot.
0021 213
```



```
0021 214 :  
0021 215 : During this memory locate and test loop, the following registers are  
0021 216 : used:  
0021 217 :  
0021 218 : R0 - the contents of the slot's configuration register;  
0021 219 : the 3rd memory controller register;  
0021 220 : R1 - bit position within memory present map;  
0021 221 : R2 - address of the next byte in the 750-specific adapter  
0021 222 : type table; address of TEST_QUAD_750 routine  
0021 223 : R3 - the default adapter type for the current slot;  
0021 224 : the number of pages in this controller  
0021 225 : R4 - address of the configuration register at the current  
0021 226 : slot position  
0021 227 : R5 - address of next byte in RPB adapter type table  
0021 228 : R7 - address of the SCB  
0021 229 : R9 - bit setting in memory present map;  
0021 230 : starting page number in this controller  
0021 231 : R10 - address of the memory description list in RPB (pagcnt & pfn)  
0021 232 : R11 - address of the RPB  
0021 233 :  
0021 234 : Initialize the RPB slot field to a zero and obtain the default adapter  
0021 235 : type for this slot. Then set up a machine check fault handler to gain  
0021 236 : control if the loop addresses a non-existent configuration register  
0021 237 : (an empty slot). Then read the slot's configuration register.  
0021 238 :  
0021 239 : Initialize R10 (RPB memory descriptor list pointer) for search loop  
0021 240 :  
30 AB 5A 00BC CB 9E 0021 241 MOVAB RPB$ MEMDSC(R11),R10 ; Set pointer to memory description list  
00005800 8F CA 0026 242 BICL #<RPB$M_MPM ! RPB$M_USEMPM ! RPB$M_FINDMEM>, -  
002E 243 RPB$L_BOOTR5(R11) ; Clear all MA780-specific boot flags  
002E 244  
002E 245  
002E 246 TRY_NEXUS_750: ; Memory locate and test loop.  
002E 247 CLR (R5)+ ; Assume nothing on slot.  
04 A7 53 82 94 0030 248 MOVZBL (R2)+,R3 ; Get default adapter type.  
00B5'CF 9E 0033 249 MOVAB DO_NEXT_750+1,4(R7) ; Set up fault handler (+1 for  
0039 250 ; handler execution on the  
0039 251 ; interrupt stack).  
50 64 D0 0039 252 MOVL (R4),R0 ; Read CR at current slot.  
003C 253  
003C 254  
003C 255 : Execution continues here if the configuration register is present.  
003C 256 : Load the adapter type into the RPB field. Then, if the adapter type  
003C 257 : is a memory controller, proceed to test memory. Otherwise, move to  
003C 258 : the next SBI slot.  
003C 259 :  
003C 260  
003C 261 TSTL R3 ; Is this a floating slot?  
003E 262 BNEQ FIXED_SLOT ; Branch if not  
FF A5 50 90 0040 263 MOVB R0,-1(R5) ; Save type read from config register  
04 11 0044 264 BRB CHECK_TYPE ; Check if memory controller  
0046 265  
0046 266 FIXED_SLOT: ; Slot is fixed assignment  
FF A5 53 90 0046 267 MOVB R3,-1(R5) ; Save fixed type  
004A 268  
004A 269 CHECK_TYPE: ; Check adapter type for memory  
004A 270
```

| PC | Op | Op2 | Op3 | Op4 | Op5 | Op6 | Op7 | Op8 | Op9 | Op10 | Op11 | Op12 | Op13 | Op14 | Op15 | Op16 | Op17 | Op18 | Op19 | Op20 | Op21 | Op22 | Op23 | Op24 | Op25 | Op26 | Op27 | Op28 | Op29 | Op30 | Op31 | Op32 | Op33 | Op34 | Op35 | Op36 | Op37 | Op38 | Op39 | Op40 | Op41 | Op42 | Op43 | Op44 | Op45 | Op46 | Op47 | Op48 | Op49 | Op50 | Op51 | Op52 | Op53 | Op54 | Op55 | Op56 | Op57 | Op58 | Op59 | Op60 | Op61 | Op62 | Op63 | Op64 | Op65 | Op66 | Op67 | Op68 | Op69 | Op70 | Op71 | Op72 | Op73 | Op74 | Op75 | Op76 | Op77 | Op78 | Op79 | Op80 | Op81 | Op82 | Op83 | Op84 | Op85 | Op86 | Op87 | Op88 | Op89 | Op90 | Op91 | Op92 | Op93 | Op94 | Op95 | Op96 | Op97 | Op98 | Op99 | Op100 | Op101 | Op102 | Op103 | Op104 | Op105 | Op106 | Op107 | Op108 | Op109 | Op110 | Op111 | Op112 | Op113 | Op114 | Op115 | Op116 | Op117 | Op118 | Op119 | Op120 | Op121 | Op122 | Op123 | Op124 | Op125 | Op126 | Op127 | Op128 | Op129 | Op130 | Op131 | Op132 | Op133 | Op134 | Op135 | Op136 | Op137 | Op138 | Op139 | Op140 | Op141 | Op142 | Op143 | Op144 | Op145 | Op146 | Op147 | Op148 | Op149 | Op150 | Op151 | Op152 | Op153 | Op154 | Op155 | Op156 | Op157 | Op158 | Op159 | Op160 | Op161 | Op162 | Op163 | Op164 | Op165 | Op166 | Op167 | Op168 | Op169 | Op170 | Op171 | Op172 | Op173 | Op174 | Op175 | Op176 | Op177 | Op178 | Op179 | Op180 | Op181 | Op182 | Op183 | Op184 | Op185 | Op186 | Op187 | Op188 | Op189 | Op190 | Op191 | Op192 | Op193 | Op194 | Op195 | Op196 | Op197 | Op198 | Op199 | Op200 | Op201 | Op202 | Op203 | Op204 | Op205 | Op206 | Op207 | Op208 | Op209 | Op210 | Op211 | Op212 | Op213 | Op214 | Op215 | Op216 | Op217 | Op218 | Op219 | Op220 | Op221 | Op222 | Op223 | Op224 | Op225 | Op226 | Op227 | Op228 | Op229 | Op230 | Op231 | Op232 | Op233 | Op234 | Op235 | Op236 | Op237 | Op238 | Op239 | Op240 | Op241 | Op242 | Op243 | Op244 | Op245 | Op246 | Op247 | Op248 | Op249 | Op250 | Op251 | Op252 | Op253 | Op254 | Op255 | Op256 | Op257 | Op258 | Op259 | Op260 | Op261 | Op262 | Op263 | Op264 | Op265 | Op266 | Op267 | Op268 | Op269 | Op270 | Op271 | Op272 | Op273 | Op274 | Op275 | Op276 | Op277 | Op278 | Op279 | Op280 | Op281 | Op282 | Op283 | Op284 | Op285 | Op286 | Op287 | Op288 | Op289 | Op290 | Op291 | Op292 | Op293 | Op294 | Op295 | Op296 | Op297 | Op298 | Op299 | Op300 | Op301 | Op302 | Op303 | Op304 | Op305 | Op306 | Op307 | Op308 | Op309 | Op310 | Op311 | Op312 | Op313 | Op314 | Op315 | Op316 | Op317 | Op318 | Op319 | Op320 | Op321 | Op322 | Op323 | Op324 | Op325 | Op326 | Op327 | Op328 | Op329 | Op330 | Op331 | Op332 | Op333 | Op334 | Op335 | Op336 | Op337 | Op338 | Op339 | Op340 | Op341 | Op342 | Op343 | Op344 | Op345 | Op346 | Op347 | Op348 | Op349 | Op350 | Op351 | Op352 | Op353 | Op354 | Op355 | Op356 | Op357 | Op358 | Op359 | Op360 | Op361 | Op362 | Op363 | Op364 | Op365 | Op366 | Op367 | Op368 | Op369 | Op370 | Op371 | Op372 | Op373 | Op374 | Op375 | Op376 | Op377 | Op378 | Op379 | Op380 | Op381 | Op382 | Op383 | Op384 | Op385 | Op386 | Op387 | Op388 | Op389 | Op390 | Op391 | Op392 | Op393 | Op394 | Op395 | Op396 | Op397 | Op398 | Op399 | Op400 | Op401 | Op402 | Op403 | Op404 | Op405 | Op406 | Op407 | Op408 | Op409 | Op410 | Op411 | Op412 | Op413 | Op414 | Op415 | Op416 | Op417 | Op418 | Op419 |
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|



```
52 00D2'CF 9E 00A7 328 MOVAB TEST_QUAD_750,R2 ; Page test routine address
    FF51' 30 00AC 329 BSBW BOOSTEST_MEM ; Test the specified range of PFN's
    04 BA 00AF 330 POPR #^M<R2> ; Retrieve fixed nexus table address.
    01 11 00B1 331 BRB DO_NEXT_750 ; Do the next controller if any
    00B3 332
    00B3 333 .ALIGN LONG ; Longword-aligned handler.
    00B4 334
    00B4 335
    00B4 336 ; Fault handler for non-existent configuration register, or unreadable
    00B4 337 ; registers, or a non-memory controller slot device. Restore stack
    00B4 338 ; pointer, clear all errors, and try for another slot if any remain.
    00B4 339
    00B4 340
    00B4 341 DO_NEXT_750: ; Skip to next slot.
    00B4 342 MOVL FP,SP ; Restore stack pointer.
    26 FFFFFFFF 5D DO 00B4 343 MTPR #-1,#PRS_MCESR ; Clear any faults.
    54 2000 C4 DA 00B7 344 MOVAB IO750$AL-PERNEX(R4),R4 ; Move to next slot.
    09 5C F5 00BE 345 SOBGTR AP,NEXT_NEXUS_750 ; If still a slot, loop.
    00C6 346
    00C6 347 ;
    00C6 348 ; Reestablish the normal machine check fault handler.
    00C6 349
    00C6 350
    04 A7 0001'CF DE 00C6 351 MOVAL UNEXP_MCHK+1,4(R7) ; Reset SCB vector.
    8A D4 00CC 352 CLRL (R10)+ ; Indicate end of RPB memory descr list
    05 00CE 353 RSB ; Return to main routine.
    00CF 354
    00CF 355 ;
    00CF 356 ; Extra label and branch here to loop back through the slot testing
    00CF 357 ; code.
    00CF 358
    00CF 359
    00CF 360 NEXT_NEXUS_750: ; Try the next slot.
    FF5C 31 00CF 361 BRW TRY_NEXUS_750 ; Branch to top of loop.
    00D2 362 ;++
    00D2 363
    00D2 364 ; Functional Description:
    00D2 365 ;
    00D2 366 ; Test a page of 780 memory.
    00D2 367 ;
    00D2 368 ; Calling Sequence:
    00D2 369 ;
    00D2 370 ; JSB TEST_QUAD_750
    00D2 371 ;
    00D2 372 ; Inputs:
    00D2 373 ;
    00D2 374 ; R0 = starting address to test
    00D2 375 ; R1 = Quad word iteration count (64)
    00D2 376 ; R11= Address of RPB
    00D2 377 ;
    00D2 378 ; Outputs:
    00D2 379 ;
    00D2 380 ; Returns via RSB if the entire page is OK
    00D2 381 ; Error exit via Machine Check code to BOO$PAGE_MCHECK
    00D2 382 ;
    00D2 383 ;--
    00D2 384
```

```

      80  60  7C  00D2  385 TEST_QUAD_750:
      80  80  D1  00D2  386          CCRQ      (R0)          ; Test 1 quadword at a time.
      00D4  387          CMPL      (R0)+,(R0)+          ; Clear a quadword.
      00D7  388          ;                               ; Read both longwords, and
      00D7  389          ;                               ; advance to next quadword.
      00D7  390          ;
      00D7  391          ; If no gross errors occur in the clear to the quadword or in the
      00D7  392          ; subsequent read instruction, then execution continues below. Otherwise
      00D7  393          ; execution goes to the fault handler.
      00D7  394          ;
      00D7  395          ;
      F8  51  F5  00D7  396          SOBGTR R1,TEST_QUAD_750          ; Continue clearing unless done.
      01  30  AB  E1  00DA  397          BBC      #RPBSV_CRDTEST,-          ; Branch if CRD test is requested.
      00DC  398          RPB$R_BOOTR5(R11),10$
      00DF  399  5$:      RSB
      00E0  400          ;
      00E0  401          ; Check if a CRD error occurred on this page.
      00E0  402          ;
      00E0  403  10$:
      7E  00F20000 9F  D0  00E0  404          MOVL      @#I0750$AL_IOBASE,-(SP) ; Get memory CSR 0.
      00F20000 9F  6E  D0  00E7  405          MOVL      (SP),@#I0750$AL_IOBASE ; Clear errors, just in case.
      8E  20000000 8F  D3  00EE  406          BITL      #^X20000000,(SP)+ ; Check for CRD error.
      E8  13  00F5  407          BEQL      5$          ; Branch if no CRD error occurred.
      0A  11  00F7  408          BRB      ERR_EXIT_750          ; Else take error path.
      00F9  409          ;
      00F9  410          .ALIGN LONG          ; All handlers longword-aligned.
      00FC  411          ;
      00FC  412          ;
      00FC  413          ; Handler that gains control when a page has gross memory errors. Just
      00FC  414          ; clear the error, recover the stack top, and advance to the next page.
      00FC  415          ;
      00FC  416          ;
      26  FFFFFFFF 8F  DA  00FC  417 PAGE_MCHECK_750:          ; Handle machine check.
      FEFA' 31  0103  418          MTPR      #-1,#PR$_MCESR          ; Clear error indicator.
      0103  419 ERR_EXIT_750:
      0106  420          BRW      BOO$PAGE_MCHECK          ; Exit to common bad page code
      0106  421          .END
```



BTMEM750  
Symbol table- Configure and Test 11/750 Memory <sup>K</sup> 815-SEP-1984 23:42:42 VAX/VMS Macro V04-00  
4-SEP-1984 23:03:06 [BOOTS.SRC]BTMEM750.MAR;1Page 10  
(3)

|                  |            |    |    |
|------------------|------------|----|----|
| ADAP TYPE 750    | 00000000   | R  | 02 |
| BOOSPAGE_MCHECK  | *****      | X  | 02 |
| BOOSTEST_MEM     | *****      | X  | 02 |
| CHECKMEM 750     | 00000010   | RG | 02 |
| CHECK_TYPE       | 0000004A   | R  | 02 |
| DMPSS-TR         | = 00000008 |    |    |
| DMPSV-TR         | = 00000018 |    |    |
| DO NEXT 750      | 000000B4   | R  | 02 |
| ERR_EXIT 750     | 00000103   | RR | 02 |
| FIXED_SLOT       | 00000046   | RR | 02 |
| INIT_SEARCH      | 00000014   | R  | 02 |
| IO750\$AL_IOBASE | = 00F20000 |    |    |
| IO750\$AL_NNEX   | = 00000010 |    |    |
| IO750\$AL_PERNEX | = 00002000 |    |    |
| NDTS_MB          | = 00000020 |    |    |
| NDTS-MEM16NI     | = 00000010 |    |    |
| NDTS-MPM0        | = 00000040 |    |    |
| NDTS-MPM1        | = 00000041 |    |    |
| NDTS-MPM2        | = 00000042 |    |    |
| NDTS-UB0         | = 00000028 |    |    |
| NDTS-UB1         | = 00000029 |    |    |
| NEXT_NEXUS 750   | 000000CF   | R  | 02 |
| PAGE_MCHECK_750  | 000000FC   | R  | 02 |
| PRS_CADR         | = 00000025 |    |    |
| PRS_MCESR        | = 00000026 |    |    |
| RPBSL_BOOTR5     | = 00000030 |    |    |
| RPBSL_MEMDSC     | = 000000BC |    |    |
| RPBSM_FINDMEM    | = 00004000 |    |    |
| RPBSM-MPM        | = 00000800 |    |    |
| RPBSM-USEMPM     | = 00001000 |    |    |
| RPBSV-CRDTEST    | = 00000010 |    |    |
| STEP_ARRAY       | 0000006B   | R  | 02 |
| TEST_QUAD 750    | 000000D2   | RR | 02 |
| TRY_NEXUS 750    | 0000002E   | RR | 02 |
| TRY_NXT_ARRAY    | 00000059   | R  | 02 |
| UNEXP_MCHK       | *****      | X  | 02 |

+-----+  
! Psect synopsis !  
+-----+

| PSECT name | Allocation       | PSECT No. | Attributes |     |     |     |     |       |       |      |       |       |      |  |  |  |  |  |
|------------|------------------|-----------|------------|-----|-----|-----|-----|-------|-------|------|-------|-------|------|--|--|--|--|--|
| . ABS      | 00000000 ( 0.)   | 00 ( 0.)  | NOPIC      | USR | CON | ABS | LCL | NOSHR | NOEXE | NORD | NOWRT | NOVEC | BYTE |  |  |  |  |  |
| \$ABSS     | 00000000 ( 0.)   | 01 ( 1.)  | NOPIC      | USR | CON | ABS | LCL | NOSHR | EXE   | RD   | WRT   | NOVEC | BYTE |  |  |  |  |  |
| YBTMEM     | 00000106 ( 262.) | 02 ( 2.)  | NOPIC      | USR | CON | REL | LCL | NOSHR | EXE   | RD   | WRT   | NOVEC | LONG |  |  |  |  |  |

+-----+  
! Performance indicators !  
+-----+

| Phase              | Page faults | CPU Time    | Elapsed Time |
|--------------------|-------------|-------------|--------------|
| Initialization     | 31          | 00:00:00.06 | 00:00:00.55  |
| Command processing | 113         | 00:00:00.67 | 00:00:02.30  |
| Pass 1             | 185         | 00:00:03.89 | 00:00:09.83  |
| Symbol table sort  | 0           | 00:00:00.37 | 00:00:00.59  |

|                        |     |             |             |
|------------------------|-----|-------------|-------------|
| Pass 2                 | 86  | 00:00:01.15 | 00:00:02.57 |
| Symbol table output    | 6   | 00:00:00.04 | 00:00:00.04 |
| Psect synopsis output  | 2   | 00:00:00.03 | 00:00:00.02 |
| Cross-reference output | 0   | 00:00:00.00 | 00:00:00.00 |
| Assembler run totals   | 425 | 00:00:06.21 | 00:00:15.91 |

The working set limit was 1350 pages.  
20780 bytes (41 pages) of virtual memory were used to buffer the intermediate code.  
There were 20 pages of symbol table space allocated to hold 317 non-local and 3 local symbols.  
421 source lines were read in Pass 1, producing 13 object records in Pass 2.  
15 pages of virtual memory were used to define 14 macros.

+-----+  
! Macro library statistics !  
+-----+

| Macro library name                  | Macros defined |
|-------------------------------------|----------------|
| -----                               | -----          |
| \$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1 | 0              |
| \$255\$DUA28:[SYS.OBJ]LIB.MLB;1     | 4              |
| \$255\$DUA28:[SYSLIB]STARLET.MLB;2  | 6              |
| TOTALS (all libraries)              | 10             |

386 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:BTMEM750/OBJ=OBJ\$:BTMEM750 MSRC\$:BTMEM750/UPDATE=(ENH\$:BTMEM750)+EXECML\$/LIB+LIB\$:BOOTS.MLB/LIB



0037 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

